Chebyshev iteration method

November 1, 2021

1 Preliminary

Chebyshev polynomial

- Recurrence definition
$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

- Trigonometric definition
$$T_n(x) = \begin{cases} \cos(n \arccos x) & \text{if } |x| \leq 1 \\ \cosh(n \operatorname{arcosh} x) & \text{if } x \geq 1 \\ (-1)^n \cosh(n \operatorname{arcosh}(-x)) & \text{if } x \leq -1 \end{cases}$$

- Minimal ∞-norm[1] . For any given $n \geq 1$, among the polynomials of degree n with leading coefficient 1 (monic polynomials),
$$f(x) = \frac{1}{2^{n-1}} T_n(x)$$
is the one of which the maximal absolute value on the interval $[-1, 1]$ is minimal.

2 Iterative solvers

Suppose we want to solve a linear system $Ax = b$ where A is a large sparse positive definite matrix. Further denote the spectrum of A: spectrum$(A) = \{\lambda_1, \dots, \lambda_n\}$, where $0 < \lambda_1 \leq \cdots \leq \lambda_n$.

A general iteraitve solver can be formulated as:

$$x_{k+1} = x_k + \alpha_{k+1}(b - Ax_k) \tag{1}$$

[1] `https://en.wikipedia.org/wiki/Chebyshev_polynomials`

, and denote b_k as $r_k = b - Ax_k$.

Let error $e_k = x - x_k$:

$$\begin{aligned} e_{k+1} &= x - x_{k+1} \\ &= (x - x_k) - (x_{k+1} - x_k) \\ &= (x - x_k) - \alpha_{k+1}(b - Ax_k) \\ &= e_k - \alpha_{k+1}A(x - x_k) \\ &= (I - \alpha_{k+1}A)e_k \end{aligned} \tag{2}$$

Then,

$$e_k = \prod_{i=1}^{k}(I - \alpha_k A)e_0 \tag{3}$$

Norm of error:

$$\|e_k\| = \prod_{i=1}^{k}\|(I - \alpha_k A)e_0\| \leq \prod_{i=1}^{k}(\|I - \alpha_k A\|) \cdot \|e_0\| \tag{4}$$

To minimize the error after k iteration $\Rightarrow$ minimize $\prod_{i=1}^{k}\|I - \alpha_k A\|$.

2.1 Richardson method

In Richardson method, all α_k's are set to a constant α: $\alpha_k \equiv \alpha$.

$$x_{k+1} = x_k + \alpha(b - Ax_k) \tag{5}$$

In this case, minimize $\prod_{i=1}^{k}\|I - \alpha_k A\| \Rightarrow$ minimize $\|I - \alpha A\| \Rightarrow$ minimize $\max_{\lambda \in \text{spectrum}(A)} |(1 - \alpha\lambda)|$.

Choosing the best α:

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \max_{\lambda \in \text{spectrum}(A)} |1 - \alpha\lambda| \tag{6}$$

$$1 - \alpha^*\lambda_1 = \alpha^*\lambda_n - 1 \Rightarrow \alpha^* = \frac{2}{\lambda_1 + \lambda_n} \tag{7}$$

2.2 Chebyshev method

Choosing the best $\alpha_i, i = 1, 2, \ldots, k$.

$$\{\alpha_i\}_{i=1,\ldots,k} = \underset{\alpha_i}{\operatorname{argmin}} \left\{ \max_{\lambda \in \text{spectrum}(A)} |(1 - \alpha_1\lambda) \cdots (1 - \alpha_k\lambda)| \right\} \tag{8}$$

Suppose that we don't know all eigenvalues of A and only know $M = \lambda_n$ and $m = \lambda_1$.

Relaxation: $\lambda \in \text{spectrum}(A) \Rightarrow \lambda \in [m, M]$.

$$\{\alpha_i\}_{i=1,\ldots,k} = \underset{\alpha_i}{\operatorname{argmin}} \left\{ \max_{\lambda \in [m,M]} |(1 - \alpha_1\lambda) \cdots (1 - \alpha_k\lambda)| \right\} \tag{9}$$

Find a degree-k polynomial $p_k(x)$ with $p_k(0) = 1$, which minimizes $\max_{\lambda \in [m,M]} |p_k(\lambda)| \Rightarrow$ The minimal ∞-norm property of Chebyshev polynomial.

Parameterize $\lambda \in [m, M]$ as $\lambda = \frac{M+m}{2} + \frac{M-m}{2} t$ and $t \in [-1, 1]$.

Optimal polynomial:

$$p_k(\lambda) = T_k \left(\frac{\lambda - (M+m)/2}{(M-m)/2} \right) \Big/ T_k \left(-\frac{M+m}{M-m} \right) = \prod_{i=1}^{k} (1 - \alpha_i t) \tag{10}$$

Corresponing α_k (zero-points of Chebyshev polynomial):

$$\alpha_i^{-1} = \frac{1}{2} \left(M + m + (M-m) \cos \frac{\pi(2i-1)}{2k} \right) \quad i = 1, 2, \ldots, k \tag{11}$$

3 Three-term iteration

We want to find the optimal polynomial which deviates least from zero in $[\lambda_{min}, \lambda_{max}]$. The answer is the scaled Chebyshev polynomial:

$$P_k(\lambda) = \frac{T_k((\lambda - d)/c)}{T_k(-d/c)} \tag{12}$$

We want to utilize the recurrence property of Chebyshev polynomial.

$$\begin{aligned}
P_{k+1}(\lambda) T_{k+1}(-d/c) &= 2\frac{\lambda - d}{c} T_k((\lambda - d)/c) - T_{k-1}((\lambda - d)/c) \\
&= 2\frac{\lambda - d}{c} P_k(\lambda) T_k \left(-\frac{d}{c} \right) - P_{k-1}(\lambda) T_{k-1} \left(-\frac{d}{c} \right) \\
&= 2\frac{\lambda - d}{c} P_k(\lambda) T_k(-\frac{d}{c}) - P_{k-1}(\lambda) \left(-2\frac{d}{c} T_k \left(-\frac{d}{c} \right) - T_{k+1} \left(-\frac{d}{c} \right) \right)
\end{aligned}$$

Reorganize items:

$$\begin{aligned}
T_{k+1} \left(-\frac{d}{c} \right) (P_{k+1}(\lambda) - P_{k-1}(\lambda)) &= 2\frac{\lambda - d}{c} P_k(\lambda) T_k \left(-\frac{d}{c} \right) + 2\frac{d}{c} P_{k-1}(\lambda) T_k \left(-\frac{d}{c} \right) \\
&= 2 T_k \left(-\frac{d}{c} \right) \left[\frac{\lambda - d}{c} P_k(\lambda) + \frac{d}{c} P_{k-1}(\lambda) \right]
\end{aligned}$$

Replace λ by matrix A, and multiply $e_0 = x - x_0$ on both sides (note that $P_k(A)e_0 = e_k = x - x_k$):

$$\begin{aligned}
T_{k+1}\left(-\frac{d}{c}\right)(x_{k+1} - x_k) &= 2T_k\left(-\frac{d}{c}\right)\left[\frac{1}{c}(A - dI)(x_k - x) + \frac{d}{c}(x_{k-1} - x)\right] \\
&= 2T_k\left(-\frac{d}{c}\right)\left[\frac{1}{c}(Ax_k - Ax) - \frac{d}{c}(x_k - x_{k-1})\right] \\
&= 2T_k\left(-\frac{d}{c}\right)\left[-\frac{1}{c}r_k - \frac{d}{c}(x_k - x_{k-1})\right] \\
&= -\frac{2}{c}T_k\left(-\frac{d}{c}\right)(r_k + d(x_k - x_{k-1}))
\end{aligned}$$

We have:

$$\begin{aligned}
x_{k+1} - x_{k-1} &= -\frac{2}{c}\frac{T_k(-d/c)}{T_{k+1}(-d/c)}(r_k + d(x_k - x_{k-1})) \\
&= \frac{2}{c}\frac{T_k(d/c)}{T_{k+1}(d/c)}(r_k + 2d(x_k - x_{k-1}))
\end{aligned}$$

$\Delta x_{k+1} = x_{k+1} - x_k$ is:

$$\begin{aligned}
x_{k+1} - x_k &= \frac{2}{c}\frac{T_k(d/c)}{T_{k+1}(d/c)}r_k + \left(\frac{2d}{c}\frac{T_k(d/c)}{T_{k+1}(d/c)} - 1\right)(x_k - x_{k-1}) \\
&= \frac{2}{c}\frac{T_k(d/c)}{T_{k+1}(d/c)}r_k + \frac{1}{T_{k+1}(d/c)}\left[\frac{2d}{c}T_k\left(\frac{d}{c}\right) - T_{k+1}\left(\frac{d}{c}\right)\right](x_k - x_{k-1}) \\
&= \frac{2}{c}\frac{T_k(d/c)}{T_{k+1}(d/c)}r_k + \frac{T_{k-1}(d/c)}{T_{k+1}(d/c)}(x_k - x_{k-1}) \\
&= \frac{2}{c}\frac{T_k(d/c)}{T_{k+1}(d/c)}r_k + \frac{T_{k-1}(d/c)}{T_{k+1}(d/c)}\Delta x_k
\end{aligned}$$

So, the general iteraiton is:

$$x_{k+1} = x_k + \alpha_{k+1}r_k + \beta_{k+1}\Delta x_k \tag{13}$$

where

$$\begin{aligned}
\alpha_k &= \frac{2}{c}\frac{T_{k-1}(d/c)}{T_k(d/c)} \\
\beta_k &= \frac{T_{k-1}(d/c)}{T_{k+1}(d/c)}
\end{aligned}$$

Initial iteration:

$$x_1 = x_0 + \Delta_0 \qquad \Delta_0 = \frac{1}{d}r_0 \tag{14}$$

In actual computation, α_k's and β_k's are generated recursively[1]:

$$
\begin{aligned}
\alpha_1 &= \frac{2d}{2d^2 - c^2}, \quad \beta_1 = d\alpha_1 - 1 \\
\alpha_k &= \left[d - (c/2)^2 \alpha_{k-1}\right]^{-1}, \quad \beta_k = d\alpha_k - 1
\end{aligned}
\tag{15}
$$

Complete algorithm[2]:

Algorithm 1 Chebyshev iteration

Require: A, b, x_0, λ_{max}, λ_{min}, $iter$, ϵ
Ensure: x

$d \leftarrow (\lambda_{max} + \lambda_{min})/2$
$c \leftarrow (\lambda_{max} - \lambda_{min})/2$
$x \leftarrow x_0$
$r \leftarrow b - Ax$
for $i = 1 : iter$ **do**
    $z \leftarrow r$
    if $i = 1$ **then**
        $p \leftarrow z$
        $\alpha \leftarrow 1/d$
    else if $i = 2$ **then**
        $\beta \leftarrow 1/2 * (c * \alpha)^2$
        $\alpha \leftarrow 1/(d - \beta/\alpha)$
        $p \leftarrow z + \beta * p$
    else
        $\beta \leftarrow (c * \alpha)^2$
        $\alpha \leftarrow 1/(d - \beta/\alpha)$
        $p \leftarrow z + \beta * p$
    end if
    $x \leftarrow x + \alpha * p$
    $r \leftarrow b - Ax$
    if $\|r\| < \epsilon$ **then**
        break
    end if
end for
return x

The update rule in the algorithm is a little different from Eq.(15), but they generate the same results (note that β's are different in algorithm and Eq.(15)).

[2] https://en.wikipedia.org/wiki/Chebyshev_iteration

References

[1] Thomas A Manteuffel. The tchebychev iteration for nonsymmetric linear systems. *Numerische Mathematik*, 28(3):307–327, 1977.

[2] Maxim A Olshanskii and Eugene E Tyrtyshnikov. *Iterative methods for linear systems: theory and applications*. SIAM, 2014.

[3] Huamin Wang. A chebyshev semi-iterative approach for accelerating projective and position-based dynamics. *ACM Trans. Graph.*, 34(6), October 2015.